UNITED STATES DISTRICT COURT
SOUTHERN DISTRICT OF NEW YORK

REALTIME DATA, LLC d/b/a IXO,	)	
Plaintiff,	)	Case No. 1:11-cv-6696-KBF 1:11-cv-6701-KBF 1:11-cv-6704-KBF
vs.	)	
MORGAN STANLEY, ET AL.,	)	JURY TRIAL DEMANDED ECF Case
Defendants.	)	

DECLARATION OF JAMES STORER IN SUPPORT OF DEFENDANTS' MOTION FOR PARTIAL SUMMARY JUDGMENT OF INVALIDITY OF THE PATENTS-IN-SUIT FOR FAILURE TO SATISFY THE DEFINITENESS AND WRITTEN DESCRIPTION REQUIREMENTS OF 35 U.S.C. § 112

I, James Storer, declare as follows:

1. I have been engaged by counsel for Defendants as an expert in connection with this litigation.

2. I understand that Plaintiff Realtime Data LLC d/b/a IXO ("Realtime") filed this lawsuit in July 2009 alleging infringement by Defendants of U.S. Patent No. 7,714,747 ("the '747 Patent"), U.S. Patent No. 7,417,568 ("the '568 Patent"), and U.S. Patent No. 7,777,651 ("the '651 Patent") (collectively "the patents-in-suit").

3. I also understand that this Declaration is being submitted in support of Defendants' Motion for Partial Summary Judgment of Invalidity of the Patents-in-Suit for Failure to Satisfy the Definiteness and Written Description Requirements of 35 U.S.C. § 112.

4. I expect to testify at any summary judgment hearing regarding the subject matters set forth in this report, if asked about these matters by Judge Katherine Forrest or by the parties' attorneys.

QUALIFICATIONS

5. I am currently a Professor in computer science at Brandeis University. I have worked in the field of computer science, including work relating to data compression, image processing, and video processing, for over thirty years.

6. I received my B.A. in Mathematics and Computer Science from Cornell University in 1975, my Masters in Electrical Engineering and Computer Science from Princeton University in 1977, and my Ph.D. in Electrical Engineering and Computer Science from Princeton University in 1979. I worked in the industry as a researcher at AT&T Bell Laboratories from 1979 to 1981 before joining the faculty of Brandeis University.

7. Between 1981 and the present, with the exception of one year during which I was a visiting professor at Harvard University, I have worked in the Computer Science Department at Brandeis University. Between 1981 and 1986, I was an Assistant Professor; between 1986 and 1992, an Associate Professor; and since 1993, I have been a full Professor. Between 1993 and 2002, I served as the Chairman of the Computer Science Department. I am currently Full Professor of Computer Science and a member of the Brandeis Center for Complex Systems.

8. I have been involved in computer science research since 1976. My research has been funded by a variety of governmental agencies, including the National Science Foundation (NSF), National Aeronautics and Space Administration (NASA), and Defense Advanced Research Projects Agency (DARPA). In addition, I have been awarded industrial grants.

9. I am a member of the ACM and the IEEE Computer Society. I routinely serve as referee for papers submitted to such journals as JACM, SICOMP, Theoretical CS, Computer Journal, J. Algorithms, Signal Processing, JPDC, Acta Informatica, Algorithmicia, IPL, IPM, Theoretical CS, J. Algorithms, Networks, IEEE J. Robotics & Automation, IEEE Trans. Information Theory, IEEE Trans. Computers, IEEE Trans. Image Processing, Proceedings of the IEEE, IBM J. of R&D, J. Computer and System Sciences. I have served as an editor for Information Processing and Management, Journal of Visual Communication and Image Representation, and the Proceedings of the IEEE.

10. I am the founder of the Annual Data Compression Conference (DCC), and have served as Conference Chair from 1991 to present. This is the leading conference devoted entirely to Data Compression. Papers accepted for presentation at DCC are subject to a selective peer review process, and many major research projects and results in the field have first been presented at DCC. A hardbound copy of the proceedings is published by the IEEE Computer Society each year.

11. I have served as guest editor for a number of professional journals, including Proceedings of the IEEE, Journal of Visual Communication and Image Representation, and Information Processing and Management. I have served as a program committee member for various conferences, including IEEE Data Compression Conference, IEEE International Symposium on Information Theory, Combinatorial Pattern Matching (CPM), International Conference on String Processing and Information Retrieval (SPIRE), Conference on Information and Knowledge Management (CIKM), Conference on Information Theory and Statistical Learning (ITSL), Sequences and Combinatorial Algorithms on Words, Dartmouth Institute for Advanced Graduate Studies Symposium (DAGS), International Conference on Language and

Automata Theory and Applications (LATA), DIMACS Workshop on Data Compression in Networks and Applications, Conference on Combinatorial Algorithms on Words.

12. My research work has included, among other areas, research relating to data compression (including text, images, and video), data archiving, content based image retrieval, image and video processing, and parallel computing. I have authored or co-authored over one hundred technical publications, which have been peer-reviewed, in these and related areas. I have published several books, including one on data compression and a textbook on computer algorithms and data structures that includes significant material on data compression. In addition, I have recently edited a book on Hyperspectral Data Compression, in which I am coauthor of one of the chapters.

13. I am also the inventor on three United States patents, all relating to data compression software and hardware (two of which I am sole inventor, and one co-inventor) and co-inventor of two patents pending.

14. A list of my qualifications is set forth in my *curriculum vitae*, a copy of which is attached as Exhibit A. I have also attached a list of my publications to my curriculum vitae.

15. For the purposes of this declaration, my opinions are based on my review of the patents-in-suit, the asserted claims, the file history of the patents-in-suit, the references cited during prosecution, dictionary definitions, and my knowledge of and experience in the field of data compression and decompression. I also reviewed the declaration submitted by Michael Ian Shamos, in support of the Plaintiff's Opening Claim Construction Brief.

BACKGROUND

16. The ’747 patent, titled “Data Compression Systems and Methods,” was filed on January 8, 2007 and issued by the PTO on May 11, 2010. The ’747 patent is a continuation of, and shares the same specification as, the withdrawn ’761 and ’506 patents. I understand that the asserted claims of the ’747 at issue in Defendants’ motion are claims 1, 7, 8 and 13.

17. The ’651 patent, titled “System and Method for Data Feed Acceleration and Encryption” was filed on June 2, 2008 and issued August 17, 2010. I understand that the asserted claims of the ’651 patent at issue in Defendants’ motion are claims 1, 4, 6, 7 and 12.

18. The patents are principally focused on data compression, not data decompression. At the time that the ’747 patent application was first filed on January 8, 2007, the inventors described the invention as follows:

> The present invention is directed to systems and methods for providing fast and efficient ***data compression using a combination of content independent data compression and content dependent data compression***.

’747 Patent, col. 3:43-46 (emphasis added). In fact, the word decompression never even appears in the “Summary of the Invention.”

19. The patents describe two alternative procedures that can be used to identify what encoders will be used to identify possible compression algorithms for encoding or compressing an uncompressed data block in the incoming data stream, and eventually determining what, if any, provides the best acceptable level of compression.

20. One procedure is used when the system is unable to recognize the type of data in the data block. (*See* ’747 Patent, col. 3:55-56, col. 4:53-54). In this event, the uncompressed and

unrecognized data block is sent to, and if possible encoded by, each of the system's "E" encoders. (*See id.* at col. 8:10-12, col. 9:57-59, col. 11:7-9).

21. The second procedure is employed "if the data type of the data block is identified" (*Id*. at col. 3:53-54, col. 4:46-47; *see also* col. 28:36-44). In this event the recognized and uncompressed data block is sent to, and if possible encoded by, appropriate ones of the system's "D" encoders. (*See id.* at col. 16:12-39).

22. According to the patents, the "D" encoders may include "any number 'n' of those lossless or lossy encoding techniques currently well known within the art such as MPEG4, various voice codecs, MPEG3, AC3, AAC, as well as lossless algorithms such as run length, Huffman, Lempel-Ziv Dictionary Compression, arithmetic coding, data compaction, and data null suppression" (*See id.* at col. 16:25-34) and the "E" encoders may include any number "of those losseless encoding techniques currently well known within the art such as run length, Huffman, Lempel-Ziv Dictionary Compression, arithmetic coding, data compaction, and data null suppression." (*Id.* at col. 16:44-47). That is, the same encoding algorithms (or compression methods) may be included in both the set of "D" encoders and the set of "E" encoders.

23. In either event, the described system then determines the level of compression that was achieved by each "D" or "E" encoder that was able to encode the data block.

24. If any encoder, "D" or "E," achieves an appropriate "threshold" level of compression, the system selects and then appends to the data block with the highest level of compression a "descriptor" that shows what compression algorithm was employed to achieve that level of compression, and outputs the data block with the attached descriptor. The

information in the appended descriptor indicates only what compression algorithm was used. It does not indicate what encoder was used, or whether a “D” or an “E” encoder was employed.

25. If, on the other hand, no encoder achieves sufficient compression, i.e., “there are no encoded data blocks having a [sufficiently high] compression ratio,” then “the original unencoded input data block is selected for output” (*See, e.g.,* ’747 Patent, col. 8:34-47). In this event a different type of descriptor, a null data compression type descriptor “that indicates that no data encoding has been applied to the input data block,” is appended to the unencoded and uncompressed data block, and “the unencoded data block with its corresponding null data compression type descriptor is then output.” (*Id*. at col. 8:41-45).

26. This procedure for deciding whether to send an uncompressed data block to either the “D” or “E” encoders, attempting to encode the block using the encoders in selected block, and then determining what levels of compression were achieved, is shown in Figures 13A and 13 B of the ’747 patent, and described in the flow chart that is Figure 14C. Figure 13A of the ’747 patent illustrates that there is a different data flow (to encoder “E”) of a data block whose type is recognized by the system than (to encoder “D”) of a data block whose type is not recognized by the system.

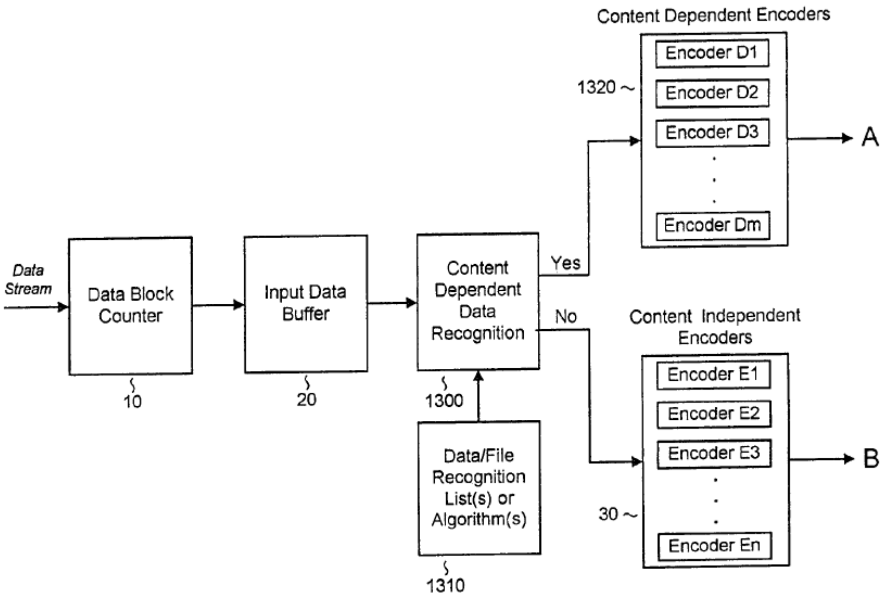


FIGURE 13A

’747 Patent, FIG. 13A.

27. Figure 13B shows that the determination of what compression ratios were obtained is made regardless of whether the data block is encoded by the content dependent “D” encoders or by the content independent “E” encoders.

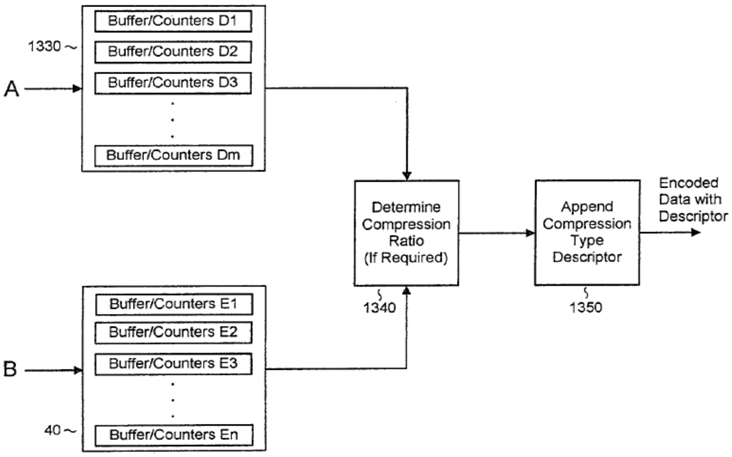


FIGURE 13B

’747 Patent, FIG. 13B.

28. What type of descriptor is appended depends on whether a desired level of compression was achieved, as shown in Figure 14C. The content of the descriptor depends only on what (if any) encoding algorithm was used. If sufficient compression is achieved, a "corresponding encoding descriptor" block 1424 is appended. If not, a null descriptor 1434 is appended.

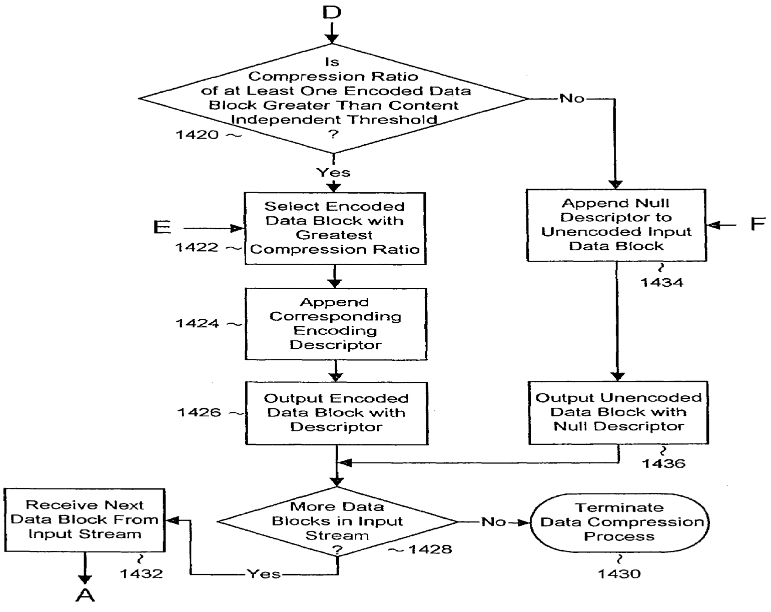

D
Is Compression Ratio of at Least One Encoded Data Block Greater Than Content Independent Threshold ?
1420
No
Yes
E
Select Encoded Data Block with Greatest Compression Ratio
1422
Append Null Descriptor to Unencoded Input Data Block
F
1434
Append Corresponding Encoding Descriptor
1424
Output Encoded Data Block with Descriptor
1426
Output Unencoded Data Block with Null Descriptor
1436
Receive Next Data Block From Input Stream
1432
A
More Data Blocks in Input Stream ?
1428
Yes
No
Terminate Data Compression Process
1430

FIGURE 14C

'747 Patent, FIG. 14C

29. Identical Figures 6 of the '651 patent and 11 of the '747 patent show a decompression system. Figure 11, and an extract from the flow chart of Figure 12, are set forth below.

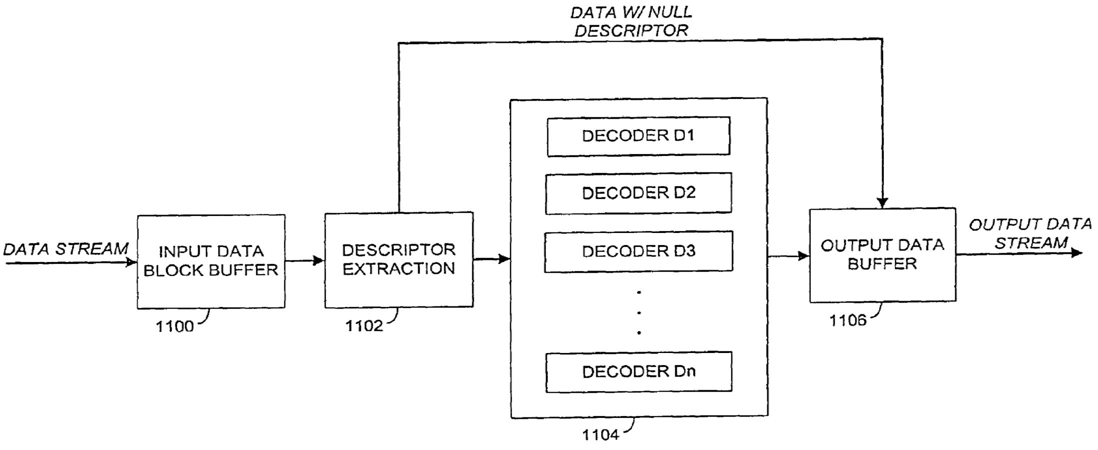
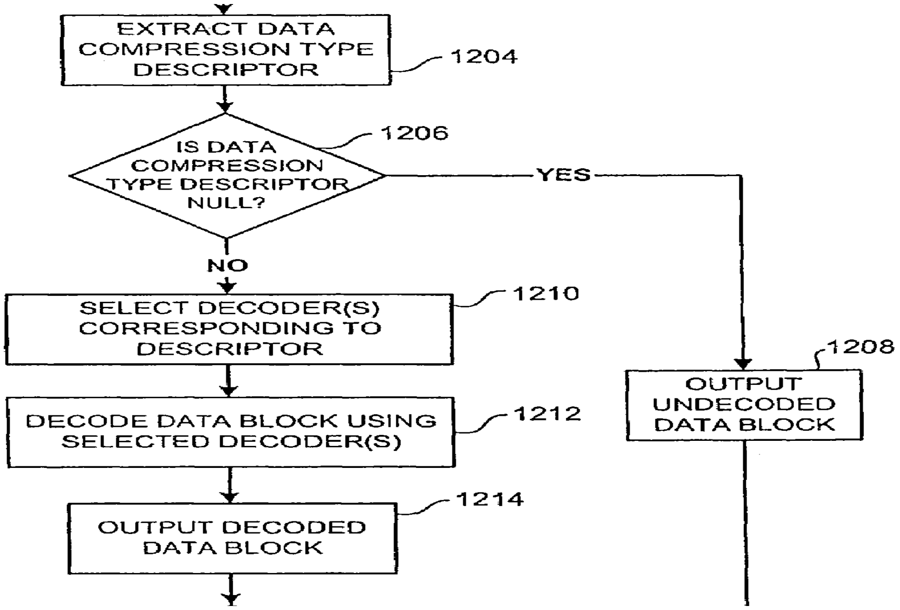


FIG. 11


EXTRACT DATA COMPRESSION TYPE DESCRIPTOR
1204
IS DATA COMPRESSION TYPE DESCRIPTOR NULL?
1206
YES
NO
SELECT DECODER(S) CORRESPONDING TO DESCRIPTOR
1210
DECODE DATA BLOCK USING SELECTED DECODER(S)
1212
OUTPUT DECODED DATA BLOCK
1214
OUTPUT UNDECODED DATA BLOCK
1208

’747 Patent, FIG. 11, 12

30. As shown in these figures, the “descriptor extraction module … receives the… input data block” and us[es] “methods known by those skilled in the art to extract the data compression type descriptor associated with the data block.” This language is not in the patent

('747 Patent, col. 14:49-54). "If the data compression type descriptor is determined to be null…, then no decoding is applied to the input data block and the original uncoded data block is output…. On the other hand, if the data compression type descriptor is determined to be any value other than null…, the corresponding decoder or decoders are then selected." (*Id*. at col. 15:22-30). As recited in the claims, this selection is "based on the descriptor." A non-null compression type descriptor indicates what algorithm/technique was used to encode the compressed data block, but does not indicate whether that particular algorithm/technique was in a "D" or "E" encoder.

31. The terms "content dependent data compression" and "content independent data compression" are not terms of art used in the data compression field.

32. An encoded data block is decoded in a way that Realtime calls the "reverse" of the encoding algorithm. If decompression is attempted using a different algorithm, the decompressed data is incomprehensible and useless. If, for example, the data block was encoded using a run length, Huffman, Lempel-Ziv Dictionary Compression, or arithmetic coding, the reverse of that algorithm must be used for decompression.

33. The patent teaches that the descriptor appended to the data block will identify which algorithm was used, but it will not indicate whether that algorithm used to encode the data block was in a "D" or "E" encoder.

34. "Content independent data decompression" or "content dependent data decompression" are not terms with known meanings to persons of skill in the art. In my over thirty years of work in the area of data compression and decompression, I have never encountered those terms outside of the asserted patents as terms of the art. I have also not

encountered the terms “content independent data compression” or “content dependent data compression” outside of the asserted patents as terms of art.

35. A person of ordinary skill in the art cannot determine the meaning of either “content independent data decompression” or “content dependent data decompression” on review of the patents specifications. The claims of the patents do not describe what “content dependent” or “content independent” data decompression technique the selected lossless decoder utilizes. Nothing in the specification teaches that the selection of a decoder depends on how the encoding algorithm was selected, or that the decoder is selected based on the particular encoder, rather than the particular algorithm, used to encode the data block.

36. As described in the patents, “Content dependent data compression” and “content independent data compression” are ***not*** compression algorithms that are actually used to compress a data packet. Rather, as discussed in the patents, these terms appear to refer to procedures that are used - in advance of any actual compression - to select algorithms that might possibly be able to compress a data packet.

37. The only place the term “content independent data decompression” is used anywhere in the specifications of either the ’747 or the ’651 patents is the statement in the former that “FIG. 11 is a block diagram of a content independent data decompression system according to one embodiments of the present invention” (’747 Patent, col. 5:45-47), and in the latter that “FIG. 6 is a diagram of a system/method for providing content independent data decompression, which may be implemented for providing accelerated data transmission according to the present invention” (’651 Patent, col. 5:66 - col. 6:2).

38. FIG. 11 and FIG. 6 show that some (but not all) of an input data stream (presumably the portion of the stream that contains compressed data) is fed to a block (1104 in the ’747 Patent and 165 in the ’651 Patent) of decoders, and (presumably) decoded data is output to an “output data buffer” and thence to the “output data stream.” They also show nothing about what technique the decoders such as run length, Huffman, Lempel-Ziv Dictionary Compression, or arithmetic coding use to decompress or decode anything.

39. Figure 12 of the ’747 patent (there is no corresponding drawing in the ‘651 patent), shows that whether and how data incoming to the decoder is decoded depends on the *descriptor* that is appended to the data, and *not* on the *content* of the compressed (or in some circumstances uncompressed) data itself. If the descriptor indicates that the data in a data block was not compressed, that data block simply bypasses the decoder. If, on the other hand, the descriptor indicates that the data in the block was compressed, the data block is decoded by selecting “decoder(s) corresponding to descriptor” (’747 Patent, FIG. 12, step 1210). According to the patent specifications, that the “descriptor may possess values corresponding to null (no encoding applied), a single applied encoding technique, or multiple encoding techniques” (’651 Patent, col. 17:23-25; ’747 Patent, col. 14:54-57).

40. The specifications of the patents describe, and other limitations in the patent claims require, that the decompression technique be selected based on the *descriptor* that is appended to the compressed data. Beyond providing that “[d]ecoding techniques are selected based on their ability to effectively decode the various different types of encoded input data generated by the data compression systems described above” (’651 Patent, col. 17:36-39; ’747 Patent, col. 14:67 - col. 15:3), neither patent teaches what these decoding techniques are, or suggest that the selection depends in any way on the *content of the compressed data*.

41. Once the data has been encoded, a user skilled in the art has essentially no choice as to what algorithm is employed to decompress. It has long been known in the field that data compression can be "reversed" using the same (or the reverse of the) algorithm that was used to compress it. Otherwise, there is no useful output. The patents do not know how to choose that decompression algorithm based on whether "content dependent" or "content independent" data compression had been "utilized" to select the compression algorithm that provided the highest level of compression.

42. Decoding "utilizing content independent data decompression" and decoding "utilizing content dependent data decompression" are mutually exclusive alternatives. The claims require that the choice depends on what the "descriptor indicates." Specifically claim 1 of the '747 Patent requires

> "decompressing the data block with a selected lossless decoder utilizing content dependent data decompression, ***if the descriptor indicates the data block is encoded utilizing content dependent data compression***"; and
>
> "decompressing the data block with a selected lossless decoder utilizing content independent data decompression, ***if the descriptor indicates the data block is encoded utilizing content independent data compression***."

'747 Patent, claim 1 (emphasis added).

43. Decompression is not independent of compression or the descriptor. As a practical matter the data must be decompressed using a decompresson algorithm or method that is specifically for output produced by the compression algorithm or method used to compress it. The patent specification, but nothing in the claim, suggests that this is so: "[d]ecoding techniques are selected based on their ability to effectively decode the various different types of

encoded input data generated by the data compression systems described above" ('651 Patent, col. 17:36-39; '747 Patent, col. 14:67 - col. 15:3).

44. The decoders that decompress compressed data never see the uncompressed data – all that is known about the "content" of the original uncompressed data is that some encoder used a compression algorithm to compress it. No uncompressed data that has been compressed by any encoder is ever sent to any decoder. A single decoder may use any one of a number of decompression techniques, i.e. decompression algorithms or methods. Similarly, a single encoder may use any one of a number of algorithms or methods to compress or encode a data block. Therefore, simply selecting an encoder does not equate to selecting a particular decompression technique. Similarly, nothing in the patent specifications teaches what technique a selected decoder uses to decode a particular compressed data block.

45. As a person skilled in the art, I cannot determine the meaning of the word "content" as used in the phrases "content independent data decompression" and "content dependent data decompression. On its face, "content independent data decompression" appears to refer to decompression that is "independent" of the "content" of the compressed data; and "content dependent data decompression" appears to refer to decompression that is "dependent" on the "content" of the compressed data. However, decompression is never "independent" of compression. To say that "content" refers to the "content" of the compressed data or the "content" of the descriptor that is transmitted with the compressed data would be inconsistent with "content *independent data decompression.*" The patent specifications do not reference or teach that decompression is based on the content of the compressed data. The word "content" also cannot mean the "content" of original uncompressed data. The decoders that *decompress*

compressed data never see the original uncompressed data, and know nothing other than that some encoder was able sufficiently to compress it.

46. A person skilled in the art would not be able to understand the meaning of the patent claims because they provide no discernible distinction between “content dependent data decompression” and “content independent data decompression.”

47. I understand that Realtime and its expert contend that both terms “content independent data decompression” and “content dependent data decompression” have the same meaning; i.e., that Realtime and its Expert say that the terms mean “revers[ing] the compression. (*See* Realtime’s Opening Claim Construction Brief (“Realtime Br.”) at 26 (emphasis in original); Shamos Decl. ¶ 12). If these two terms mean what Plaintiff and its expert say they do, namely that “applying a reverse technique - i.e., decompression - to reverse the compression of data achieved through dependent [or independent] data compression”, then the last claim limitation, and everything in the next-to-last limitation starting with “utilizing,” are entirely superfluous.

48. The remaining portion of the claim would still require “decompressing the data block with a selected lossless decoder.” It has long been known that any successful decompression “reverses” compression Realtime calls “reversing” the original compression is inherent in “decompressing the data block.” Realtime and its Expert admit that “the decompression algorithm is the reverse of whatever compression algorithm was used to encode the data.” (Realtime Br., 26; *see* Shamos Decl. ¶ 13).

I declare under the penalty of perjury under the laws of the United States that the foregoing is true and correct.

Executed this 4th day of April 2012 in Waltham, MA.

James Storer